Niveau(x) d'enseignement: 5 Numéro EDUQ : 8270451
Centre d'intérêt: 64

Montpetit, Guy.

Quelques critères pour une décision sur le choix d'un langage de programmation pour l'enseignement : résultats tirés d'une expérience de recherche au Collège Edouard-Montpetit... / Guy Montpetit. -- Longueuil : Le Collège ; [Québec] : Ministère de l'éducation, Service de la recherche du secteur collégial, 1973.
13 f. ; 28 cm.

"Recherche subventionnée par le Ministère de l'éducation du Québec (Service de la recherche du secteur collégial) et réalisée grâce à la coopération du Conseil national de la recherche, Ottawa".

Cote Dewey: 371.3'9445/M799q Cote LC: LB/1028.5

Descripteur(s): *Enseignement par ordinateur *Difficulté d'apprentissage *Secondaire (Niveau) *Collégial (Niveau)

Identificateur(s) de nom: *LOGO

Identificateur(s) de sujet: *Langage de programmation

Suite à des expériences d'initiation à l'ordinateur réalisées au cégep Edouard-Montpetit, le document a pour objet de faire la description et l'analyse des principaux problèmes intellectuels rencontrés par les participants. Il vise aussi à établir les principaux critères d'ordre pédagogique concernant le choix d'un langage de programmation pour l'enseignement.

Au total, 21 étudiants de première année du collégial et 36 professeurs du secondaire et du collégial ont participé à l'expérimentation. Ils ont suivi des cours d'utilisation de l'ordinateur à raison de trois heures par semaine durant une session pour les étudiants et à raison d'une heure par semaine durant six semaines pour les professeurs. Tous avaient accès à un ordinateur muni du langage LOGO à l'aide duquel ils devaient réaliser un certain nombre de programmes.

L'auteur, qui a donné les cours, a remarqué certaines résistances de la pensée qui nuisent au travail de l'étudiant. Elles se situent au niveau de la représentation des objectifs, de la traduction d'un objectif en moyens pour obtenir un résultat, enfin au niveau de l'identification des objets du travail. Le langage de programmation idéal pour fins d'enseignement doit, selon lui, être naturel, clair, puissant et bénéficier d'importantes capacités d'édition.

Type de document: r
Région(s) administrative(s): 062
Financement: Bénéficiaire - Collège Edouard-Montpetit
Organisme(s) source(s) - MEQ - DGEC

Programme(s) de subvention - SRD
Auteur de l'analyse: fr

E D U Q 8270451

COLLEGE EDOUARD-MONTPETIT
SERVICE DE LA RECHERCHE (Secteur collégial)
MINISTERE DE L'EDUCATION DU QUEBEC

APR 1973
RECU
DIGEC

Mars 1973
Guy Montpetit

QUELQUES CRITERES POUR UNE DECISION SUR LE CHOIX D'UN LANGAGE DE PROGRAMMATION POUR L'ENSEIGNEMENT

(Résultats tirés d'une expérience de recherche au Collège Edouard-Montpetit Longueuil, Canada)

Recherche subventionnée par le Ministère de l'Education du Québec, (Service de la recherche du secteur collégial) et réalisée grâce à la coopération du Conseil National de la Recherche, Ottawa.

PROLOGUE

Nous écrivons cet article en pensant à ceux qui ont la responsabilité de décider des différents moyens pédagogiques qui seront mis à la disposition des élèves. L'avenir des relations homme-machine est particulièrement intéressant pour la pédagogie. En effet la vie intellectuelle de plusieurs millions d'enfants dépend en partie de la qualité des instruments de travail qui leur seront offerts. Nous voudrions proposer une série de criteres d'ordre pédagogiques -tirés d'une expérience intensive de l'utilisation de l'ordinateur- pouvant s'appliquer au choix d'une technologie contemporaine en pédagogie.

Introduction:

Ce texte sera une étude détaillée de cinq expériences intensives d'utilisation de l'ordinateur dans l'enseignement. Trois des expériences ont été faites avec des étudiants de Collège (1ère année) et deux courtes expériences avec des enseignants du niveau secondaire et collégial. Elle nous ont permis d'identifier avec certitude les difficultés de toute démarche intellectuelle. Nous les retrouvons quelles que soient les variables de l'expérience en cause.

Après avoir situé les objectifs qui ont déterminé le contenu de ces expériences nous ferons la description et l'analyse des principaux problèmes intellectuels rencontrés par les étudiants de tout âge et toute culture.

Enfin et c'est là l'essentiel de ce texte nous voudrions établir des critères d'ordre pédagogiques portant sur le choix d'un langage de programmation pour l'enseignement. Nous tenterons de sortir le débat de ce qui semble habituellement un "cadre d'informaticiens" pour le situer sur le terrain d'une clientèle étudiante. Lui redonner en quelque sorte sa perspective dynamique.

Il est inutile de construire des systemes idéaux si les personnes à qui ils sont destinés ne sont pas atteints, où encore qu'il faille supposer qu'ils ne sont pas assez intelligents pour nos systèmes.

I - DESCRIPTION DE L'EXPERIENCE

Il convient de décrire en quelques mots la situation des étudiants qui ont participé aux expériences qui serviront de base à notre discussion.

1. CADRE THEORIQUE: UNE THEORIE DE L'APPRENTISSAGE

Les idées pédagogiques de base qui ont déterminé le contenu de nos expériences remontent à Dewey et à Piaget. Nous croyons comme Dewey et Piaget qu'une chose s'apprend "en la faisant" et non en "l'apprenant".

Nous avons ainsi choisi une série de projets[1] réalisables par les étudiants qui avaient accès à un ordinateur muni du langage LOGO. Le lecteur intéressé par le contenu de ces expériences consultera avec satisfaction des différents mémos publiés à cet effet par le laboratoire d'Intelligence artificielle du Massachussetts Institute of Technology.

L'aspect plus dynamique de cette théorie nous indique que l'action seule ne suffit pas. On apprend dans la mesure où l'on peut réfléchir à son action. L'activité principale de la recherche consistait à établir avec les étudiants un langage qui puisse s'appliquer à la planification et au déroulement d'un projet. Identifier les "bugs" de la pensée qui nous font dévier d'une route tracée ou d'un objectif fixé, leur donner un nom pour pouvoir en parler telle était la démarche quotidienne de chaque étudiant. Pour ce faire,

---

[1] PAPERT, S. & SOLOMON, C.: Vingt projets à réaliser avec un ordinateur, juin 1971. Traduction française de Guy Montpetit.

Twenty things to do with a computer, June 1971, Mass. Institute of Technology, (AI - Memo 247) (Logo memo 2)

il faut apprendre à diviser une difficulté et c'est sans doute la meilleure méthode pour commencer à réfléchir à quelque chose. Ces principes ont dirigé l'ensemble de notre travail

2. CADRE PRATIQUE: a) Technologie disponible.

Les étudiants avaient accès en direct à l'ordinateur PDP 10 du Conseil National de la Recherche a Ottawa, qui dispose du langage LOGO. Quatre terminaux alphanumériques de type télétype étaient les seuls appareils périphériques disponibles. (Nous nous proposons d'ajouter une série d'autres appareils périphériques). Chaque étudiant pouvait donc programmer lui-même l'ordinateur en direct et en recevoir une réponse immédiate.

b) Organisation du travail.

Les étudiants recevaient ce type d'enseignement par groupe de trois à raison de trois heures semaines pendant une session, depuis le début de l'expérience 21 étudiants ont participé à une expérience intensive et 36 professeurs environ ont participé à une expérience d'une heure semaine pendant 6 semaines. Les professeurs cependant venaient parfois en groupe de 10. L'expérimentateur était le seul enseignant dans les deux cas.

3. OBJECTIFS

Les objectifs sont à la fois d'ordre technologiques et pédagogiques. Un rapport de nature technique sera soumis par Simon Curry & Lorne H. Bouchard sur les aspects technologiques en cause.

Les objectifs pédagogiques consistaient à établir une description dynamique des difficultés intellectuelles inhérentes à la réalisation des projets. Nous employons le terme dynamique pour souligner le fait que ces difficultés ne sont pas innées et sont le fruit d'une interaction entre le professeur, l'étudiant et la machine.

L'objectif final étant finalement de permettre à l'étudiant de se libérer de ces accrochages intellectuels, de les identifier lorsqu'ils se présentent dans d'autres projets, et d'engager chez lui une série d'activités qui lui permettra de réaliser des projets de complexité croissante.

II - LES RESISTANCES DE LA PENSEE DANS LA REALISATION DES PROJETS

Dans les pages qui suivent nous décrirons certaines résistances de la pensée qui empêchent l'étudiant de poursuivre un travail, ou de continuer à s'intéresser activement à la solution d'une difficulté.

Le pédagogue averti ne trouvera là rien de très nouveau. Nous croyons cependant utile de fournir les détails d'une situation privilégiée ou nous pouvons observer ces difficultés, et où surtout nous pouvons y remédier. D'autre part, comme cette expérience a servi de base à nos conclusions, nous invitons le lecteur à considérer ce chapitre à titre de référence seulement, et à passer directement au chapitre suivant qui tentera d'établir des critères pour le choix d'instruments pédagogiques.

1) La représentation d'un objectif

L'épistémologie nous apprend que pour connaître quelque chose il nous faut déjà avoir une petite idée de la réponse, ou,de ce que l'on cherche. C'est ainsi que nous demandons à un étudiant de toujours illustrer d'abord le résultat qu'il cherche à obtenir. Nous lui demandons de faire "comme si" tout était parfait et d'anticiper la situation qui se produirait si les données du problème qu'il veut résoudre étaient déjà résolues[1].

Eh bien la premiere grande difficulté d'un étudiant est qu'il refuse d'admettre qu'à un moment donné, il "réussira à vaincre les difficultés et qu'il peut ILLUSTRER le résultat qu'il veut obtenir. Tout se passe comme si il disait:

"Comment vais-je faire pour faire ça? ("Ca" étant le résultat désiré) Si je trouve comment je peux faire, je parviendrai peut-être à le faire".

---

[1] Une pédagogie basée sur la notion de PROJET à réaliser est supérieure à ce point à une pédagogie basée sur la mémorisation d'information ou sur le "DRILL". Par exemple demander à la TORTUE de dessiner un coeur, est une chose en apparence simple à représenter.

A la différence bien sûr, que ne sachant pas encore ce qu'il veut faire il s'engage dans la description des moyens qui produisent des résultats bien différents de ce qu'il souhaitait.

Cette confusion entre l'activité d'illustrer un résultat, par l'activité d'illustrer les moyens pour obtenir un résultat est particulièrement évidente dans les projets du type QUIZZ. La forme initiale d'un QUIZZ consiste simplement a demander aux étudiants de faire un programme qui pose une question dont ils doivent déterminer la nature, d'obtenir une réponse, de l'évaluer et de faire les commentaires qu'ils jugent appropriés.

Fidèle à notre démarche nous insistons pour que les étudiants ILLUSTRENT d'abord le résultat de l'éventuelle exécution de ce programme comme si il était déja défini et mis en mémoire de l'ordinateur. Afin d'appuyer notre démarche nous suggérons aux étudiants par une démonstration au tableau, de diviser leur feuille de travail en deux que nous nommons ainsi:

DEFINITION OU PROCEDURE PRODUISANT LE RESULTAT	RESULTAT DESIRE
?	?QUIZZ (TITRE DU PROJET) Qu'elle est la couleur du cheval blanc de Napoléon? GRIS IDIOT! Qu'elle est la coule r du cheval blanc de Napoléon? BLANC TU ES UN GENIE!

Fig. 1. ILLUSTRATION D'UN RESULTAT DESIRE, dans ce cas faire un programme QUIZZ qui pose une question, va chercher une réponse, et fais les commentaires appropriés.

Malgré celà, nous avons pu constater que 100% des étudiants rencontrent à un moment ou un autre cette difficulté. Il faut les"entraîner" de façon systématique à se fournir des moyens d'ILLUSTRER les différents résultats qu'ils cherchent, et de se donner un langage pour les distinguer clairement des moyens produisant ces résultats.

L'exemple le plus intéressant de cette confusion dans les QUIZZ est celui de l'étudiant qui pense immédiatement à la façon dont son programme va évaluer "la bonne réponse" à la question posée et qui en vient même à oublier de mettre dans son programme ce qui permettrait de répondre à cette question. Si au lieu d'essayer de faire son programme immédiatement en pensant à ses propres problèmes il avait anticipé le déroulement du programme, il n'aurait pas probablement pas oublié que si on posait une question il fallair aussi écouter la réponse.

L'autre forme de ce même bug se retrouve chez ceux qui pensent à obtenir une réponse du monde extérieur mais n'en tiennent pas compte et leur programme se déroule comme s'il n'y avait rien eu.

Enfin ce bug place aussi l'étudiant dans la situation cocasse de voir un collègue qui veut essayer son programme, et qui après avoir donné la réponse juste à la question posée, être la victime des commentaires souvent plus ou moins élogieux, se revoit poser la même question à nouveau.

Donc les difficultés d'illustration d'un résultat entraînent une planification du projet davantage en fonction des difficultés du sujet qu'en fonction des difficultés objectives du projet et de ses objectifs. Appliqué à l'enseignement des maths ce raisonnement pourrait peut être expliquer pourquoi les étudiants ne parviennent pas à se représenter l'idée de toute cette entreprise.

2) La traduction d'un OBJECTIF en MOYENS pour obtenir un résultat

Une fois parvenu à l'illustration à peu près complète d'un résultat désiré, l'étudiant semble cette fois souffrir du problème inverse. Il semble incapable de le diviser en parties opérationnelles. Ainsi lorsque nous voulons traduire un résultat en action, il faut essayer de trouver qu'elle est l'action principale qui produit ce résultat; s'agit-il d'une action qui se poursuit subissant des modifications de façon reconnaissable, i.e., à chaque unité de temps par exemble, quel est le sens de ces modifications, quand se préparent-elles, et quand arrêteront-elles?

Voilà une série de questions qui doivent être résolues pour diviser un résultat de façon organique et réunir les moyens qui permettront sa réalisation. Pour chacune de ces questions on rencontre chez l'étudiant une série de bugs qui ont tous la caractéristique de vouloir régler le résultat d'un seul coup. C'est l'absence de représentation du temps qui ne supporte pas l'action envisagée. Ils ne sont pas "habitués" à décrire un processus en termes temporels.

Un exemple frappant de cette difficulté se retrouve dans la description de l'action "centrale" d'un processus. Elle est souvent confondue avec le résultat final lui-même. Ainsi dans un projet simple de dessins par télétype lorsqu'on demande à un étudiant de dire qu'elle est l'action centrale du programme qui produirait le résultat suivant:

DEFINITION	RESULTAT
?POUR TRIANGLE	?TRIANGLE "XXXXX"
	XXXXX
	XXXX
	XXX
	XX
	X
	?

Fig. 2. Illustration du résultat d'un programme appelé TRIANGLE qui ayant reçu un INPUT de 5 caractères, produit la figure suivante.

On obtient systématiquement comme réponse de la part de l'étudiant que l'action centrale consiste à "faire un triangle" ou à "en enlever toujours un" alors qu'il est plus important de constater que l'action du programme est d'IMPRIMER l'objet qu'il a reçu. (Certains l'appellent INPUT) Cette action aura une durée, elle se poursuivra et on constate que l'INPUT sera modifié systématiquement et que l'action s'arrêtera alors qu'il n'y aura plus de caractères dans l'INPUT. Il faut encore ici beaucoup d'exercices et beaucoup de modèles pour entraîner les étudiants à se donner des définitions de plus en plus puissantes pour l'identification de l'action inductrice d'un processus.

Nous croyons que cette difficulté découle d'une autre plus fondamentale: LE "NAMING" que nous allons décrire maintenant.

3. LE "NAMING"(1) ET LES OBJETS

Les étudiants n'acquierrent pas naturellement l'idée d'identifier les objets qu'ils auront besoin de manipuler et de leur donner un nom afin de pouvoir en parler. Leur tendance naturelle à sauter au résultat les fait au contraire oublier d'isoler chaque variable nécessaire à l'action; ils agissent à la façon d'une locomotive en pensant que si le train part il entraînera tout le monde. Sauf que la locomotive ne connaît pas quels wagons elle doit tirer! On remarque une résistance exceptionnelle à trouver des noms et de les relier aux choses. La difficulté la plus fréquente est de confondre un nom et une chose.

Molière avait déjà remarqueé la facheuse habitude qu'ont les médecins de mettre un nom sur une chose en pensant que le problème se règlerait

---

(1) Nous employons ici le participe présent anglais pour représenter l'activité de donner un nom à une chose.

de lui-même. La confusion entre le nom et la chose est OMNIPRESENTE dans le déroulement d'un projet, tant dans sa phase de planification que d'exécution. Tous les étudiants sans exception rencontrent cette difficulté à des degrés divers.

Plus précisément il y a la difficulté spécifique de trouver un "bon nom" pour une chose. Une idée que nous considérons néfaste pour la pédagogie est couramment répandue et exprimée face à ce problème. Elle consiste à affirmer qu'un nom n'est qu'un nom et que n'importe quel nom peut être choisi. Quoique vraie dans certaines circonstances - particulièrement après qu'une chose eut-été apprise - certains noms sont de meilleurs supports représentatifs que d'autres. On peut même dire que le choix des noms par l'étudiant est le meilleur indice de compréhension que le professeur puisse observer en cours d'apprentissage. On pourrait multiplier les exemples ou un "naming" inadéquat a provoqué des difficultés d'apprentissage. Certains étudiants ont une difficulté à simplement nommer une chose. Ils bloquent, ils hésitent, et s'arrangent pour que le professeur trouve un nom pour eux.

Cette activité, rappelons-le, est reliée à la manipulation des OBJETS nécessaires à une action. La seconde difficulté spécifique relative au "naming" consiste à nommer des objets inutiles; d'identifier les mauvais acteurs, ceux qui n'ont pas une fonction spécifique. Par exemple dans l'apprentissage du jonglage la tendance est générale à vouloir donner un nom aux balles (A B C par exemple) comme si chaque balle avait une fonction spécifique. Les mains cependant ont une fonction à remplir et elles sont presque toujours nommées spontanément "gauche" et "droite" alors qu'il est facile de constater que l'on puisse trouver de meilleurs noms pour cette activité. En effet pour jongler (avec 3 balles) lorsqu'une balle redescend vers une main, cette même main devra lancer la balle qu'elle a déjà en main et attraper la balle qui descend. De sorte que la main qui lance est la même que celle qui reçoit et le nom "ACTIVE" est plus justifié que gauche ou droite.

Lorsque les étudiants parviennent à ce genre de représentation on observe un progrès décisif dans l'apprentissage.

4. La réduction au cas le plus simple

Nous touchons maintenant a une difficulté qui se manifeste sous plusieurs aspects. Savoir attaquer un problème n'est pas une chose qui est apprise dans nos curriculum actuels. Par exemple les étudiants ont l'habitude de s'attaquer aux cas très particuliers des problèmes, précisément ceux qui causent la difficulté. Ils s'accrochent a cette difficulté et ils se bloquent dans leur raisonnement. Commencer par résoudre tous les cas simples et ensuite s'attaquer au problème plus complexe nécessite également un entrainement. Par exemple dans un programme qui cherche à déterminer si telle chose est membre d'une liste d'autres choses, le cas le plus simple est précisément à cherche à ramener le membre en question en tête de liste: Le premier membre de la liste semble toujours plus facile à trouver.

```
POUR MEMBRE :CHOSE :LISTE
10 SI :LISTE :VIDE EMETS "FAUX
20 TESTE :CHOSE  PREMIER :LISTE
30 SI VRAI  EMETS  "VRAI
40 MEMBRE :CHOSE  SAUFPREMIER :LISTE
FIN
```

L'étudiant a facilement la croyance que les chose sont très compliquées avant de les avoir faites. Nous n'insisterons jamais assez sur l'idée qu'un problème doit pouvoir devenir simple au point d'en avoir l'air ridicule.

5. L'habitude de la bonne Réponse

Habitués sans doute à répondre à des questions d'examens où la réponse doit être bonne, l'étudiant ne prend pas trop de risque à essayer ses idées. Il doit absolument avoir la bonne réponse et il tolère difficilement de faire des "erreurs". Ils acceptent assez facilement les formes d'autoritarismes qui les rassurent sur la "vérité". Par exemple comment enseigner le fait que la tortue doit tourner de 360° si elle veut revenir dans la même direction. Doit-on simplement dire que c'est 360° et pourquoi.

(1) Une étudiants me disait qu'elle préférait les maths parce que là au moins elle était sûre de ne pas avoir de bug. "Si tu fais comme le prof a dit, t'es correct".

III - CRITERES SUR LE CHOIX D'UN LANGAGE DE PROGRAMMATION

De ces quelques exemples que nous ne valons pas considérer comme exhautifs, nous voudrions proposer des critères à ceux qui décideront des langages de programmation pour l'enseignement. Je m'empresse de souligner que la liste des langages disponibles n'est pas si longue et que nous ne voulons pas situer le débat sur le plan des langages spécialisés pour des projets spécialisés. Lorsque nous pouvons utiliser l'ordinateur dans l'enseignement nous croyons qu'il puisse s'appliquer à un ensemble de projets et qu'il doit d'abord être envisagé en fonction des difficultés intellectuelles de ceux qui les utilisent. Je suis presque assuré que certains d'entre-vous se disent déjà qu'"on le voit venir". C'est très bien comme celà.

LE PREMIER CRITERE:

<u>Le langage doit être naturel</u>

La première difficulté que nous avons souligné est la difficulté de se représenter un objectif. Nous croyons qu'il est nécessaire de ne pas faire intervenir trop de règles grammaticales exotiques ou formalisantes lorsqu'il s'agit de traduire un objectif en action. Le cas idéal et souhaité par tous serait celui où l'on pourrait parler à l'ordinateur comme à un ami et qui comprendrait assez bien ce qu'on lui dit. Il pourrait pas surcroît être sensible aux inconsistances de l'interlocuteur.

En principe, un observateur neutre et ignorant devrait être capable de lire un programme et de comprendre son fonctionnement. Il n'est pas nécessaire de faire intervenir des formes de traduction de sa pensée qui soient trop éloignées de la pensée de celui qui travaille.

CRITERE SUIVANT:

<u>Le langage doit être puissant</u>

Quel est le genre d'objets que l'on voudrait utiliser pour représenter notre univers. La premiere idée que nous semble importante est que l'on puisse manipuler simplement les objects non-numériques. En effet la seconde difficulté des étudiants fait référence a la traduction des objectifs en moyens. Dans plusieurs cas il est clair que l'on aurait profit à utiliser des opérations linguistiques plutôt que des opérations arithmétiques.

Par exemple:

```
POUR RENVERSER :MOT
10 SI :MOT = : "   EMETS : "
20 EMETS MOT DE DERNIER: MOT ET
             RENVERSER DE SAUFDERNIER: MOT
FIN
```

L'opérateur MOT qui a besoin de deux inputs est général et s'applique à tous les cas.
A la limite le langage pourrait traiter ses propres programmes comme des textes.

AUTRE CRITERE:

La clarté

Le langage doit pouvoir offrir des programmes limpides, lisibles. On doit pouvoir les débugger a vue d'oeil; en tout cas le professeur doit pouvoir lire sur le champ le programme d'un étudiant. En d'autre terme, le langage doit pouvoir permettre d'écrire et d'anticiper des sous-procédures. Par exemple, au moment de la définition d'un programme il ne devrait pas y avoir de messages d'erreur. Ce n'est pas le cas au moment de l'exécution.

CRITERE IMPORTANT:

Le "NAMING"

Le langage doit permettre à l'étudiant de nommer ses procédures et de nommer les variables de ces procédures. De sorte qu'il soit possible d'utiliser la récursion et de parler des objets qu'on ne connaît pas encore. Nous jugeons ce critère essentiel, sans autre détails. Il ne semble pas d'ailleurs soulever d'objections.

CRITERE A LA MODE:

Puissantes capacités d'édition

Lorsqu'un langage est perçu comme un texte il devient nécessaire de faire des éditions au texte et ceci doit être très facile afin de ne pas décourager

l'étudiant à poursuivre un projet. On doit pouvoir manipuler le texte très facilement et dans toutes les directions.

CRITERES PHILOSOPHIQUES.

1) La "permissibilité"

Tout ce qui est susceptible de créer le sentiment chez l'étudiant qu'il doit demander une permission avant d'agir est néfaste au travail. Ceci s'adresse même à la façon d'entrer en contact avec l'ordinateur. Faire disparaître les mots de passe et toutes ces petites complications inutiles d'adressage qui n'intéressent pas l'étudiant. La liberté d'inventer ses propres fonctions; un langage ayant une liste trop longue de primitifs peut avoir l'air savant.

2) Comment il transmet sa "philosophie"

Les messages d'erreurs sont souvent condescendants et surtout non-intelligibles par celui qui faisait l'erreur.

Les projets qu'il rend accessibles à l'étudiant peuvent donner l'impression qu'un "computeur" c'est une chose très sérieuse et qu'on ne doit pas s'abandonner à des choses trop éloignées de la "computation" classique.

CONCLUSION

Il ne s'agit pas d'une discussion statique et définitive. Les langages vont évoluer. Mais tant qu'ils laisseront l'opinion publique dans l'idée qu'il faut "connaître les mathématiques pour faire ça" on peut penser que les ordinateurs continueront à coûter cher. Ces machines doivent devenir accessibles car elles sont l'occasion de véhiculer des idées plus puissantes que la règle et le crayon et la mémoire humaine.

FIN